In the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously Presented) A pipelined data processor operating in a plurality of pipeline phases including at least an instruction decode pipeline phase and an execution pipeline phase capable of predicated instruction execution dependent upon the state of an instruction designated predicate register comprising:

a data register file including a plurality of read/write, general purpose data registers;

an instruction decode unit operative during said instruction decode pipeline phase receiving fetched instructions and determining the identity of at least one source operand data register, a destination operand data register and one of a plurality of functional units for execution of each instruction, said instruction decode unit further identifying a predicate register responsive to receipt of a predicated instruction;

a scoreboard bit corresponding to each data register capable of serving as a predicate register, each scoreboard bit connected to said instruction decode unit to be set to a first digital state upon determining said corresponding data register is a destination for an instruction and connected to said plurality of functional units to be reset to a second digital state opposite to said first digital state upon functional unit write of a result to said corresponding data register;

said plurality of functional units operative during an execution pipeline phase connected to said instruction decode unit for performing a data processing operation on at least one source operand recalled from at least one corresponding instruction designated source data register and producing a result, said functional unit

responsive to an instruction not a predicate instruction to write said result to an instruction designated destination data register, and

responsive to a predicate instruction to write said result to an instruction designated destination data register if said corresponding predicate data register has a first state during said execution pipeline phase regardless of said state of said corresponding scoreboard bit and to nullify said instruction and not write said result if said predicate register has a second state opposite to said first state during said execution pipeline phase regardless of said state of said corresponding scoreboard bit; and

each functional unit is further operative responsive to a predicate instruction during said instruction decode pipeline phase to nullify said predicate instruction of a following execution phase by operating at a reduced power state relative to normal instruction operation if said predicate register has said second state during said instruction decode pipeline phase and said corresponding scoreboard bit has said second state during said instruction decode pipeline phase.

2. (Original) The pipelined data processor of claim 1, wherein:

said functional unit is further operative to reset said scoreboard bit to said second digital state upon nullification of said instruction designating a corresponding data register as a destination operand data register.

3. (Canceled)

4. (Previously Presented) A method of operating a pipelined data processor operating in a plurality of pipeline phases

including at least an instruction decode pipeline phase and an execution pipeline phase capable of predicated instruction execution dependent upon the state of an instruction designated predicate register comprising the steps of:

setting a scoreboard bit to a first digital state upon determining a corresponding data register is a destination for an instruction;

resetting a scoreboard bit to a second digital state opposite to said first digital state upon a write of a result to said corresponding data register;

performing a data processing operation via a corresponding functional unit on at least one source operand recalled from at least one corresponding instruction designated source data register, producing a result and writing said result to an instruction designated destination data register in response to an instruction not a predicate instruction;

performing a data processing operation via a corresponding functional unit on at least one source operand recalled from at least one corresponding instruction designated source data register and producing a result in response to a predicate instruction designating a corresponding predicate data register and writing said result to an instruction designated destination data register regardless of said state of said corresponding scoreboard bit if said corresponding predicate data register has a first state during said execution pipeline phase;

nullifying a data processing operation of a predicate instruction by not writing said result to the instruction designated destination data register via said corresponding functional unit regardless of said state of said corresponding scoreboard bit if said corresponding predicate register has a second state opposite to said first state during said execution pipeline phase; and

nullifying a predicate instruction for a following execution phase by operating said corresponding functional unit at a reduced power state relative to normal instruction operation if said corresponding predicate register has said second state during a prior instruction decode pipeline phase and said corresponding scoreboard bit has said second state during a prior instruction decode pipeline phase.

5. (Original) The method of claim 4, further comprising the step of:

resetting a scoreboard bit to a second digital state upon nullification of said instruction designating said corresponding data register as a destination operand data register.

6. (Canceled)

7. (Previously Presented) The method of claim 4 further comprising the steps of:

statically scheduling instruction execution via a compiler; and

scheduling via said compiler a last write to a data register before an instruction decode pipeline phase of a predicate instruction designating said data register as a predicate register.

8. (Previously Presented) The pipelined data processor of claim 1, wherein:

each functional unit is operable at said reduced power state by not fetching at least one instruction operand and not toggling a corresponding register read port during said following execution phase.

9. (Previously Presented) The pipelined data processor of claim 1, wherein:

each functional unit is operable at said reduced power state by not powering said functional unit during said following execution phase.

10. (Previously Presented) The method of claim 4, wherein:

said step of operating said corresponding functional unit at a reduced power state includes not fetching at least one instruction operand and not toggling a correspond register read port during said following execution phase.

11. (Previously Presented) The method of claim 4, wherein:

said step of operating said corresponding functional unit at a reduced power state relative includes not powering said functional unit during said following execution phase.

12. (Previously Presented) The pipelined data processor of claim 1, wherein:

said scoreboard bit corresponding to each data register capable of serving as a predicate register includes

an OR gate having a first input receiving a signal from a corresponding functional unit indicating when a write instruction to said corresponding predicate register commits, a second input receiving a signal from said corresponding functional unit indicating said write instruction to said corresponding predicate register nullifies and an output, and

a flip-flop having a set input receiving an input from said instruction decode unit indicating when said corresponding predicate register is a destination for said write instruction, a reset input connected to said output of

said OR gate and a Q output indicating a state of said scoreboard bit.

13. (Previously Presented) The pipelined data processor of claim 1, wherein:

each of said plurality of functional units includes

a compare to zero unit receiving data from said instruction identified predicate register generating a signal when data stored in said instruction identified predicate register is zero,

a first AND gate having a first input receiving said output of said compare to zero unit, a second input receiving a signal indicating when said predicated instruction is in said instruction decode pipeline phase, an inverting input receiving said state of said corresponding scoreboard bit and an output, and

a second AND gate having a first input receiving said output of said compare to zero unit, a second input receiving a signal indicating when said predicated instruction is in said execution pipeline phase;

each of said plurality of functional units operable to

nullify said predicate instruction of a following execution phase by operating at a reduced power state relative to normal instruction operation upon generation of a signal at said output of said first AND gate, and

not write said result of said predicate instruction of a current execution phase upon generation of a signal at said output of said second AND gate.